PCT

(51) International Patent Classification 6 :		(11) International Publication Number:	**WO 99/29099**
H04N	A2	(43) International Publication Date:	10 June 1999 (10.06.99)

|	
(21) International Application Number: PCT/US98/25351	(81) Designated States: AT, AU, BR, CA, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, ID, IL, JP, KR, MX, NO, NZ, PL, RU, SE, SG, UA, US, VN, YU, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date: 30 November 1998 (30.11.98)	
(30) Priority Data: 08/969,338 28 November 1997 (28.11.97) US	
	Published *Without international search report and to be republished upon receipt of that report.*
(71)(72) Applicant and Inventor: ROSSIDES, Michael, T. [US/US]; 11167 E. Mirasol Circle, Scottsdale, AZ 85259 (US).	

(54) Title: SYSTEMS FOR USING PRICES TO SCREEN E-MAIL
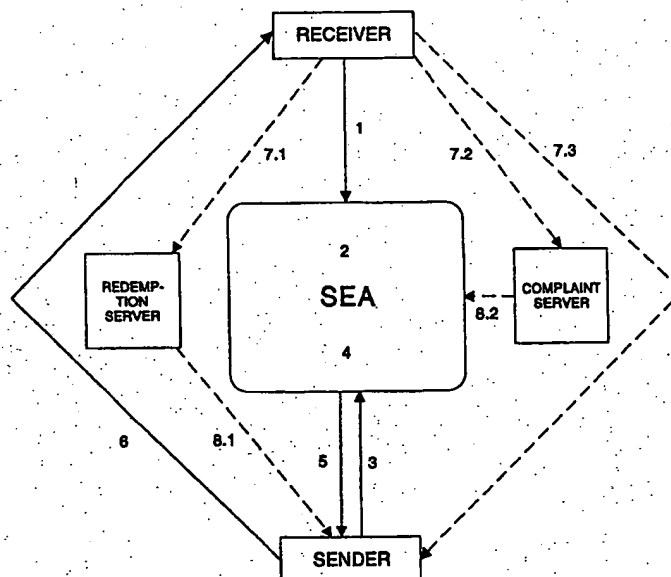
(57) Abstract

The invention is a novel directory for enabling users to request e-mail literature using monetary signals. The second invention can include means for estimating the probability that a recipient will reply to e-mail ads. The second invention can be combined with a system for affixing payment to the e-mail and with a system for redeeming payments affixed to e-mail.

Systems for Using Prices to Screen E-mail

Summary of the Invention

The invention is a novel directory for enabling users to request e-mail literature using monetary signals. The second invention can include means for estimating the probability that a recipient will reply to e-mail ads. The second invention can be combined with a system for affixing payment to the e-mail and with a system for redeeming payments affixed to e-mail.

Brief Description of the Drawings

Figure 1 illustrates the operation of an online database system for posting monetary requests for e-mail literature.

Figure 2 illustrates the operation of an online database system for posting monetary requests for e-mail literature and a combined mailserver for sending e-mail literature.

Figure 3 illustrates the operation of an online database system for posting monetary requests for e-mail literature and combined mailserver for sending e-mail literature and for affixing and redeeming micropayments.

DESCRIPTION OF THE INVENTION

Introduction

E-mail lends itself to abuse because, through automation, the cost of sending messages to millions of addresses is negligible. Therefore, mass e-mail (spam) can lead to a deluge of junk mail for receivers. However, if e-mail includes micropayments to receivers then e-mail is not free and many problems associated with spam disappear.

An *expected* micropayment can be affixed to an e-mail message. It is "affixed" in the sense that it is a message—more specifically, a contract— that is added to the main message of the e-mail.

Tis specification describes means for screening e-mail accroding to the amopunt of money affixed to it.

This specification also describes a new kind of database system that enables people to use monetary signals (prices) to request e-mail literature (especially sales literature). In addition, this directory can be enhanced with statistical means for evaluating the probability that someone who places a request for literature will end up replying to that literature or end up buying something described in that literature.

Modular Approach

This specification takes a modular approach in the sense that it describes modular systems that can be put together. Modules can be combined to form larger systems, there is no best combination of modules. Further, the modules do not have to be combined into a single system. The modules can be considered independent inventions. Moreover, multiple ways exist to achieve the objectives of these modules. Therefore, we cover several basic ways without committing to a single method.

E-stamps and L-stamps Defined

An *e-stamp* is a payment made by the sender of an e-mail to the recipient. Payments (sender-pays-receiver stamps) can affixed to e-mail in a wide variety of ways. One way is to affix an electronic lottery ticket to an e-mail. Such expected micropayments affixed to e-mail will be called *l-stamps*. The "l" stands for "lottery" because l-stamps are equivalent to lottery tickets with a specified expected value.

2

Some Definitions

In this section we give some definitions that will be used throughout this specification. Some of these will be elaborated on as the discussion progresses. New definitions will be added as needed.

E-mail. Also called *an e-mail message.* For our purposes, an e-mail message will mean information that is sent from one computer to another and that is seen by senders and receivers and that "arrives" in an e-mail box. Thus, for our purposes, this information includes what is known as the header and the body and includes attached files but does not include the what is conventionally called "the envelope" (information that is not normally seen by e-mail users).

Header. The part of an e-mail that is seen initially when an e-mail arrives in a receiver's e-mail box. The header may be made up of several fields, including but not restricted to: sender, receiver and subject fields.

Body. The content of an e-mail that must be "opened" by a receiver. In other words, the content that is not seen initially when an e-mail arrives in the receiver's e-mail box. Files attached to an e-mail, including audio and graphics files, will be considered part of the body.

L-stamp. A lottery ticket, with a specified expected payment value, affixed to an e-mail. Also referred to as a *stamp.*

EV. The expected payment value of an L-stamp.

Payoff. The amount of money that can be won in an L-stamp bet.

RN. A random number that is used to decide an L-stamp bet.

RNS. An RN supplier. Also called an *RNG* (RN Generator).

L-mail. E-mail that has an L-stamp affixed to it. By "affix" we mean that L-stamp information is *added* to an e-mail message. With physical mail the common term for adding a stamp is to say that a stamp is "affixed". We adopt that usage but also use *add* as a synonym for *affix.*

Server. A computer—processor, memory, input/output means, and programming means—connected to a network.

3

L-stamp server (LS). A server with programming means for sending e-mails and affixing L-stamps to those e-mails. Also called an *L-mail server*.

Redemption server (RS). A server with programming means for redeeming L-stamps.

Full L-stamp system (FLS). A system that combines the functions of an LS and an RS. In other words, a computer that both sends out e-mails with L-stamps and that enables people to get paid for winning stamps.

Sender. The party responsible for sending an L-mail. Also referred to as *Seneca*.

Payer. The party responsible for paying off a winning L-stamp. (Usually, we will assume that the sender is also the payer, although the payer can be the agent of the sender.)

Receiver. The party that receives an L-mail. Also called the *recipient*. Also referred to as *Reece*. Note: An L-stamp will usually be made out to an addressee rather than to a person's real name. Although not always valid, the assumption is that the person who accesses an e-mail box is the person who has the rights to any L-stamp sent to that box.

E-mail Receiving Program: The program or set of programs that make up a receiver's mail client and/or mail user agent. In other words, the programs that process a receiver's incoming and outgoing e-mail. Also called the *receiver's e-mail program*.

Procedure. A series of steps executed by a computer.

What Is an L-stamp Server?

An LS is a computer that includes programming means for adding to an e-mail a set of information defining an L-stamp. The information affixed depends on the kind of L-stamp that is involved. Thus, an LS is not one machine but a type of machine. Particular LS's will differ depending on the kind of L-stamps they affix. Again, physical lottery tickets provide a direct analogy. The machines that produce these tickets vary depending on the kind of tickets involved. Likewise, the machines that affix L-stamps will vary depending on the kind of L-stamps involved.

4

(The term LS can refer to a computer that is on a network but does not serve the network in the typical sense of a "server". An LS can refer to anything from a PC to a mainframe. It is the functional ability to add an L-stamp to an e-mail that defines an LS.)

What Is a Redemption Server?

Like an LS, an RS is not one machine but a type of machine. RS's will vary depending on the kind of L-stamps they redeem. It is the functional ability to redeem an L-stamp that defines an RS.

Ambiguity in the Term E-mail

In this specification we describe how an LS adds L-stamp information to an e-mail. As discussed, we will think of an e-mail message as the header and the body content that receivers see. When we add the L-stamp information to either, we will sometimes refer to the header or body as the *modified header* or *modified body*.

At what point does the e-mail become an L-mail? There is no clear dividing line. Even when all the L-stamp information is added to an e-mail, thereby making it an L-mail, we can still call it an e-mail that has been stamped with certain payment information. Because more than one piece of information defines an L-stamp, it is sometimes easier to visualize an e-mail with pieces L-stamp information added to it rather than think of an L-mail. So we often refer to an e-mail that has L-stamp information affixed to it. Note, then, that there is sometimes ambiguity when the term *e-mail* is used. It is left the reader to interpret the meaning from the context.

The Reason for Defining and Naming Files

In this specification we define and name several kinds of files. In giving these definitions and names, we are not trying to say exactly how L-stamp systems would store information. Our goal is just to explain the kinds of information that L-stamp systems would store. Those skilled in the art will easily see alternative ways of referring to this information.

5

Chapter 1
Monetary Methods and Systems for Sorting, Screening, Requesting and Sending E-mail Literature

The purpose of L-stamps is to enable senders of e-mail to pay receivers for receiving those e-mails. These payments in turn enable e-mail to be used as an advertising medium by mass e-mailers. Without such payments, mass e-mail would be impractical—mailboxes would be flooded.

The existence of L-stamps—or any practical reverse postage scheme—also raises the possibility that receivers can sort and/or screen e-mails according to monetary value. Put another way, L-stamps can, along with other mechanisms, enable people to charge admission to their mailboxes.

This idea can be implemented in many ways. We see the same thing in the physical world where people have devised numerous systems and processes for charging for admission. Just consider the variety of means employed by nightclubs, airlines and amusement parks—these means include many kinds of ID's, ID checking, physical barriers, reservations systems, price posting, ticketing, ticket presenting, and payment handling.

So too, in the world of e-mail we can employ a variety of systems and processes for enabling people to charge for admission to mailboxes. These systems rely on sorting and/or screening procedures but also include price posting, ticketing of sorts and payment handling. This chapter describes such systems and processes.

3.1   Sorting E-mail According to It Monetary Value

If e-mails can contain L-stamps, Reece's e-mail program can include means for sorting her e-mails from highest EV to lowest.

As discussed in section 1.11 on extra headers, Reece's e-mail program can also include means for sorting her e-mail according to various monetary headers. For example, her program can sort e-

mails according to their EV per word or per K byte. Her program can also sort e-mails into separate folders depending on whether they have L-stamps or not.

But sorting e-mail by EV information has shortcomings in a world where mass e-mailing is allowed. In such a world, Reece's mailbox may be filled with thousands of e-mails per day, overwhelming her capacity to even scan it effectively. A solution is to enable Reece to screen her e-mails as well according to the value of their L-stamps.

3.2   Screening E-mail According to Monetary Value

Screening e-mail can mean various things. Here we will think of it as the process of preventing Reece from getting e-mails that do not fit certain conditions. Actually, this definition isn't very good so we rely on the reader's understanding of the term "screening e-mail."

We differentiate screening from sorting in the sense that sorting puts e-mails in a certain order while screening can block a recipient from seeing an e-mail altogether. An e-mail program can both sort and screen, of course, and the combination may be referred to as filtering. Normally an e-mail is screened according to certain header information it contains.

E-mail can be screened at its source by a sending program that checks whether an e-mail to a given address meets pre-specified conditions for that address. It can also be screened at its destination. Reece's e-mail program (or the mailserver that serves her e-mail) can delete offending e-mails or bounce them back to their senders.

In this section we discuss how e-mails can be screened according to the EV of their L-stamps but that does not preclude screening according to other characteristics as well.

General Admission Price

To screen e-mail according to it's EV Reece (or whoever is managing Reece's mailbox) must set an admission price, which we call a *general admission price* (GAP) or *GAP screen.*

7

A GAP screen means that any e-mail sent to the recipient must include an L-stamp with an EV greater than or equal to the GAP. Otherwise it will be blocked either at its source or destination.

The simplest GAP is one that applies to all e-mails and that specifies only that the e-mails contain L-stamps greater than or equal to a threshold. But GAP's can be created to include additional conditions. And GAP's can be differentiated to apply to different kinds of e-mails. Thus, a person can set more than one GAP. Below are some examples of useful GAP's that are more specific than a GAP that applies to all e-mails.

1) A GAP can be set to apply to a particular type of e-mail format. For example, Reece can set one GAP for graphics files and one for text files.

2) A GAP can state a price per word of text in an e-mail. To get through such a GAP screen, an e-mail must include an L-stamp whose EV divided by the number of words in the e-mail is greater than or equal to the GAP. Similarly, a GAP can be set that specifies a price per K byte of an e-mail. Such GAP's encourage parties to send short e-mail messages.

3) A GAP may include a condition that an L-stamp must include a specified credential guaranteeing its EV—guaranteeing that the L-stamp sender will pay off, that is, if the stamp is a winner.

4) A GAP can state that an L-mail must include an L-stamp that does not force the recipient to open the L-mail in order to cash the stamp.

5) A two-tier GAP can be set that states two prices: one for how much Reece expects just for receiving the L-mail and one for how much Reece expects for opening the L-mail. Thus an L-mail must include two different types of L-stamps to get through such a GAP screen (or include one L-stamp that has an EV greater than or equal to the sum of both prices and that does not require Reece to open the L-mail in order to get paid).

So, any GAP screen includes:
a) a receiver's e-mail address,
b) a price—an amount of money that a sender of an e-mail must pay the receiver for receiving the e-mail,
c) a set of conditions that stipulate the form of the e-mails that the price applies to (the price may apply to e-mails of all forms).

8

The conditions are objective in the sense that they can be implemented in a computer program that checks whether the conditions have been met. For example, a condition might stipulate that an e-mail has to be less than a certain number of K bytes. A condition cannot be subjective, as in stipulating that an e-mail must be mellifluous.

Using GAP's

As discussed, GAP's can be enforced by e-mail receiving programs that screen out L-mail that does not contain L-stamps of high enough value. Alternatively, GAP's can be enforced as a matter of policy by senders who agree to send someone an L-mail only if it contains an L-stamp with an EV greater than or equal to that person's GAP.

Either way, the sending or the receiving program must include means for entering the GAP conditions and then checking e-mails being sent or received to see if the e-mails have met the conditions. Any e-mail that fails to meet the conditions is not sent or is deleted or is sent back.

Note: A GAP Can Also Be Useful When a Receiver Simply Sorts E-mail

We have defined a GAP as a screen but we can also use it in a looser sense and apply it to situations where receivers sort their e-mails but do not screen them. In this case a GAP would refer to the idea that the receiver would not usually look at an e-mail that did not have an L-stamp greater than or equal to the GAP. Reece might sort all e-mails without a high enough value L-stamp to a separate folder or she might just manually delete all L-mails that did not contain a high enough value L-stamp. So, if Reece were to post a GAP, senders would know that she would not look at any e-mail that had an L-stamp with an EV less than that GAP.

A GAP List and a GAP List Server

Theoretically GAP's can be implemented without a GAP list—a directory of people's GAP's. In theory, a sender can deploy an agent that "knocks on people's e-mail addresses" and asks their e-mail programs what their GAP conditions are. Alternatively, Seneca can send Reece an e-mail with no L-stamp. Reece's receiving program can bounce back a message telling what her GAP is.

9

Seneca's receiving program can then capture her GAP from the bounce back e-mail. Seneca can then send L-mail's to those e-mail programs (to those addresses) whose GAP conditions are satisfactory.

It seems that a more efficient way to implement GAP's, at least for mass e-mailers, is through a central directory where people post their GAP's. We will call this directory a *GAP list* (GAPL)—a list of e-mail addresses and corresponding GAP's.

A GAPL must be maintained on a server, which we will call a *GAP list server* (GAPLS). Receiver's send their GAP's to the server and sender's access GAP's through the server. Thus, a GAPLS includes means for entering, storing and outputting GAP's and corresponding addresses. In other words, the GAPLS is an on-line directory that is made up of the GAP listings that receivers enter. Receiver can change their listings (which are password protected). Sender's rights to the listings can vary.

Source Screening Using a GAPLS

An LS can screen e-mails according to GAP's. The LS may incorporate a GAPLS, otherwise, the LS must get the GAP's from the GAPLS.

The simplest form of screening at the source is for the LS to "ask" the GAPLS to output the addresses and GAP's of receiver's that have GAP's below a certain threshold. The LS can then use these addresses as the basis of an L-mail list. Further, an L-mail list can include a GAP field as part of each record in the list.

When sending L-mails, the LS can include a rule for affixing an L-stamp equal to each GAP. That way Seneca does not pay any receiver more than that receiver's GAP.

As an example of GAP screening at the source, let us assume that Seneca is selling a telephone service by e-mail ads and that his criteria for sending an ad is how much he must pay in reverse postage. Say he sets a limit of .$10. The LS enables him to enter this limit. Then the LS gets a mailing list and gets the GAP's of the names on the list. The LS then sends L-mails only to those receiver's with GAP's less than or equal to $.10.

10

3.3 Admission Credits

GAP screens have an obvious problem: they block valuable e-mail sent by from parties who don't want to pay Reece. Sorting without screens has a similar problem: useful e-mail can be lost in a flood of unsolicited e-mail.

These problems can be handled easily where friends are concerned: Reece can agree not to cash her friends' L-stamps. Therefore, her friends can affix L-stamps with high EV's that will put their L-mails at the top of Reece's in-box.

But not everyone who has useful information to send is a friend. For example, what happens when the government wants to send Reece an e-mail, or when Reece requests information from a company?

If Reece sorts or screens e-mails according to EV, she needs to be able to give preference to e-mails from certain parties and to e-mails about certain subjects. What is needed is a guest list system of sorts.

But, since we are putting money on e-mails, the idea of a guest list is not quite right. The idea is to enable people to prioritize e-mail according to how much money is affixed to it. So, a way Reece can give priority is to grant *admission credits*. A credit enables someone to send an L-mail that has an L-stamp of high enough EV to get through Reece's GAP screen, and be sorted to the top part of Reece's list of incoming e-mails.

Credits be given to designated parties and, more than that, they can be granted for designated subjects. Reece can use these credits to tell advertisers what information she wants. In fact, a system for posting credits can be a new reverse advertising medium in the sense that potential buyers use it to request e-mail sales literature from sellers.

Admission Credit Defined

Before we can describe such a system, we need to define what we mean by an admission credit.

11

An admission credit is a right granted by the owner of a mailbox. It is the right to send that owner a pseudo-payment affixed to an e-mail. The pseudo-payment is treated as a real payment for the purposes of sorting and screening e-mail but not for the purposes of converting it into real money. A credit, then, is made up of the following set of information:

1) The grantor's e-mail address.

2) Who the credit is offered to.
a) A credit may be offered to a particular sender.
b) A credit may be offered to all senders who send an e-mail about a specified subject. In this case, the subject can be specified in free form, natural language text (rather than by pre-specified "check-box's").

3) The amount of the pseudo-payment. (We'll refer to this amount as the *credit amount.*)

4) A boilerplate clause in which the owner agrees to treat the pseudo-payment as an actual payment for the purposes of sorting and screening the e-mail by the value of any affixed stamps, *provided that the e-mail is from the specified sender or is about the specified subject.* (If the subject of the e-mail does not match the specified subject then the sender the credit might stipulate penalties, such as actual payment of the credit amount.)

5) A boilerplate clause in which the owner agrees that the pseudo-payment is not an actual payment that can be converted into legal tender.

6) A boilerplate clause stating that the credit is revocable.

7) An expiration date.

8) A signature or other authentication means such as a PIN (authentication means are not necessarily essential, see 3.45).

When we say a credit we will have in mind a grant defined by the set of information above. The grant is given by a receiver (Reece), also called an owner of a mailbox. (Note: in practice, the boilerplate clauses can be abbreviated by a single symbol.)

12

When a credit specifies a particular party—when it is "made out to" a particular party—we will call it a *guest credit*. The idea is that the credit is for a particular person, like a guest pass or personal invitation.

When the credit specifies the subject that an e-mail is to be about we will call it a *request credit*. The idea is that the receiver uses the credit to request information about a particular subject.

Publicizing Credits

For a credit to be used, a mailbox owner must communicate it to a sender or senders. Credits can be publicized on a public server or they can be communicated personally from a receiver to a sender. We discuss these matters in Section 3.4. First, let us discuss how a sender might use a credit.

Using Admission Credits

To make use of a credit, Seneca affixes the credit to an L-mail. The key part of a credit is the amount of the pseudo-payment. The credit can be affixed by itself or with an L-stamp.

The credit will stand alone if it is the full "payment" to Reece. In this case, Seneca adds, say, "$1" to the e-mail subject header. Now, he can also affix other information identifying the credit, if he thinks that is necessary. It may not be necessary to add any extra information. For example, if Seneca is Reece's friend, he might not feel the need to do so.

But, with commercial e-mail we will assume that additional information identifying the credit will be added to the e-mail. This additional information can be put in the body of the e-mail or can be added as further header information. In fact, just as L-stamps can have a separate header, a header can be created for credit amounts.

Now, if a credit is being added to an L-stamp amount then Seneca can simply add the two amounts to arrive at a *total payment* to Reece. The total payment is used for purposes of screening and sorting her L-mail. But, part of it is a pseudo-payment and part is a real payment. How then is this combined payment presented? It can be presented in various ways, all equivalent. For

13

example, the total payment can be the first figure in a subject header while the credit amount can be in parentheses.

Net Prices and Bonuses: A Credit Amount Relative to a GAP

We say that the difference between Reece's GAP and a credit amount is the *net price*. For example, if the GAP is $1 and the credit amount is $.50 then the net price of admission to Reece's mailbox is $.50, for the party or subject of the credit grant. If the difference is between the GAP and a credit amount is zero or negative then we say that the net price is zero. Further, if the difference is negative—if the credit amount is grater than the GAP—we call the difference a *bonus* or a *bonus amount*.

Discounts Instead of Credits

Now, it is obvious that a credit can be looked upon as a discount. Thus, Reece could grant a discount from her GAP for designated senders and for e-mails about designated topics. For example, if Reece's GAP is $1, she could grant a discount of $.50 rather than a credit of $.50.

A discount is defined in much the same way as a credit, except that a sender is granted the right to pay a given amount less than the GAP—rather than granted the right to make a pseudo-payment.

The set of information defining a discount is the same as a credit in that:
1) the discount is granted by an mailbox owner,
2) it is offered to a particular sender or to all senders of e-mails about a specified subject,
3) it has a denomination,
4) it has an expiration date and,
5) it has authentication information.
The difference is in the conditions of the grant. A discount includes a clause stating that the grantor agrees that the sender can pay less than the GAP for the purposes of screening e-mail. The amount of the discount is called the *discount amount*. The amount that the sender has to pay is called the *discount price*. (The discount price is equivalent to the *net price*.)

In practice, discounts may be used rather than credits. They seem easier to use. That's because under a credit method, if the net price is greater than zero, a credit and an L-stamp are need to get

14

past a GAP screen, while under a discount method, only an L-stamp is needed to get past the screen.

But credits have a couple of key advantages. One, unlike a discount, a credit can be granted that is greater than a GAP, resulting in a bonus that can be useful for sorting mail preferentially. For example, if Reece's GAP is $1, Reece can grant a credit of $2, meaning that an e-mail with such a credit will carry a bonus of $1. With this bonus, the e-mail may then be sorted to the top of Reece's list of incoming e-mails. But Reece can only grant a discount of $1, meaning that the e-mail can get through Reece's GAP screen, but once past that the e-mail will not be treated preferentially. The second advantage of credits is that they lead to much easier screening on Reece's side. It is easier to set up and use an e-mail receiving program that screens according to some GAP rather than a program that must make discount exceptions for all kinds of subjects and senders. Under a credit system, Reece could set up a GAP screen that screened out all e-mails that did not carry a total payment of high enough value. Under a discount system, subjects have to be screened, whereas with a credit method the only thing that has to be checked is the total payment amount of the L-stamp and/or credit that is affixed.

Because credits are more versatile than discounts, we will focus on them in the rest of this specification. However, we note that most of the points we make apply to discounts as well. Moreover, the discussion of credits we will focus almost exclusively on *request* credits because they are the basis of a new advertising medium/directory described in section 3.4.

Additional Request Credit Conditions

Recall that the purpose of request credits is to solicit sales literature. So, in addition to describing the subject of the literature, a request credit can state a number of other conditions that help Reece screen e-mail, and that help sellers reduce the sending of wasteful e-mails. Below is a partial list of standard conditions that a request credit can include. These conditions include instructions for senders and instructions for the server that stores the requests.

Before listing these conditions we should note that a request as described in this section is an abstract entity made up of: a set of information defining a grant, a set of conditions governing the use of the grant, and a set of information describing literature that is wanted.

15

The next section describes an online database system for request credit information. This system makes a request a more concrete entity, one that can be found and viewed by the public. This system stores this information in request files. So, when we say that a credit can include a *separate*, standard condition or descriptive clause we mean that this database system enables Reece to enter the condition or clause as a distinct part which has its own field in a request file, and which, for output and search purposes, is treated as a distinct part of a request.

Number of Times the Credit Can Be Used. A grantor can stipulate that a credit can only be used by each sender a certain number of times before it expires. This condition can be enforced by Reece who can complain when Seneca uses a credit too many times. She may complain to an agency set up for that purpose. Alternatively, if a third party is acting as a mailing agent for Reece and Seneca then the third party can observe the condition.

Privacy Requirement. A grantor can stipulate that her e-mail address is not to be given to the sellers who would send the requested literature but only to a third party entrusted with mailing the literature. In fact, we will assume that the requirement of privacy is the default choice.

Erase Requirement. A grantor can stipulate that the entire record of the credit is to be erased at a certain date. For example, someone who issues a credit encouraging drug companies to send information about anti-herpes drugs might want that fact erased forever.

Postal Code. A credit can include a zip code as a separate piece of information. A zip code is a useful specifier where literature is sought about products or services, such as restaurants, that are provided by local sellers. (A zip code can, of course, be included in the subject description but separating it out can make it easier to find by sellers.)

E-mail Format. A credit can include a condition specifying the format of the e-mail. In other words, a credit can state that the e-mail should include graphics files or audio files or multi-media files.

Product or Service Price. A credit can include the desired price range of a product or service. As with the zip code, this information can be put in the subject description but is more easily accessed if separate.

16

<u>Type of Guarantee</u>. A credit can specify that a particular type of guarantee must back up the product or service that information is sought about. This message, obviously, tells sellers what kind of guarantee they must offer.

<u>Type of Credentials</u>. A credit can specify the type of credentials that a company must have in order to send sales literature. Reece may not want to receive literature from every potential seller. She might just want a select group of sellers to send her information. This group can be identified by a credential of some sort. For example, Reece might only want Board Certified doctors to send her literature about medical treatments.

<u>Delivery Time</u>. A credit can specify a time by which Reece needs a product or service delivered.

Buyer Identity Information

A credit can include a part where Reece describes herself (she may be an organization) in some detail. This information can help Seneca and, if characterized in a standard, "check box" way, can help the sea develop statistics concerning Reece's buying behavior (see Chapter 4).

Question and Elaboration Fields

A request can also include distinct fields in which Reece directs a question or questions to sellers. The seller can then answer the question in his sales literature. A question field is not a condition that the requested literature must conform to but rather an effort to learn something specific about the subject that Reece is requesting information about. For example, if Reece is requesting information about Hotel rooms in Amalfi, she might also ask: *Do you have an Internet connection?*

A request can also include an elaboration field in which Reece elaborates upon the request as stated in the subject field (and in the other fields as well). In the elaboration field, Reece can explain in more detail what she wants. The other fields are meant to be searched by automated means. A question and an elaboration are for human examination primarily (although they can be searched as well). A person can look at these to divine better what Reece wants. Taking *Amalfi hotels* as a subject, Reece might then enter the following in the elaboration field: *I want a cozy room with a*

17

fireplace that has a window that opens out onto the sea, and that is not near a lot of traffic, and that is in a small inn or bed and breakfast rather than a large hotel. I want to stay two weeks.

Of course, Reece can enter this description in the subject field instead. The subject field has space for a free form text description that can be arbitrarily long. However, an alternative is to have a shorter description in the subject field and a longer description in an elaboration field.

Statements of Buying Intent

A request can also include statements of intent by the grantor such as the following:

"I Want to Buy". A request credit might mean that the grantor wants to buy something but not always. Thus, the credit can include a separate clause for a standard "I want to buy" message. This message alerts sellers that the grantor is an especially "live prospect."

"I Want a Price Quote". A request credit can include a separate clause for a standard "I want a price quote" message. This message, obviously, tells a sender to include a price in the requested literature.

"I'm Taking the Lowest Price". A request credit can include a separate clause for a standard "I'm taking the lowest price" message. This message, obviously, tells a sender that they buyer is price shopping.

Commitments. A request credit can include a separate clause in which the grantor commits to buying if certain conditions are met. A commitment might be needed to elicit sales literature that is costly to produce. Below we develop this point by describing on one particular kind of commitment that can be an important part of a request.

The Importance of Commitments, Particularly Those With Strike Prices

The cost of responding to a request from a prospective buyer can be prohibitive, especially when there are large numbers of roughly equivalent sellers, and especially when responding requires even a small amount of person's time. The chance of success drops such that the expected profit is less than the cost of responding. For example, say Reece is a lobbyist in Washington who enters a

18

request for information about how much it will cost to print 10,000 brochures on KromeKote, 20lb. stock, accordion folded, in 2 colors, black and blue, with camera ready art. Now, over a hundred printers in the Washington DC area could respond to this request. But it might take each 15 minutes to put together a quote. Say the time cost is $10 and the chances of winning the job are 1%. Then the profit on the job must be over $1000 in order for the expected profit to be greater than the $10 cost of responding. In this case, responding is a losing bet.

In general, a seller needs to reduce the uncertainty about the competition in order to judge whether investing in a response is a good bet. A solution to the problem is for Reece to make a *strike price commitment*. By that we mean that she precisely describes what she wants and she includes a price. The commitment is that she promises to buy at that price from the first seller who fulfills her conditions. Of course, such a commitment can be modified so that Reece commits to buy from one of the first x sellers to fulfill her conditions. By seeing the number of sellers he will be up against, Seneca can better evaluate whether it is worth it to invest in a response.
Thus, a request can include a field for making a strike price commitment.

3.4   The Sea of Requests and Its Functions

Just like GAP's, credits must be maintained on a public (online) server if they are to be most useful. We will call the server where credit information is stored by the name *the sea of requests* (the sea). We use that name because the sea is supposed to be massive list of requests entered by potential buyers (mailbox owners) to be searched (fished in) by sellers.

(Note on terminology: we will use the terms *request, request credit,* and *credit* as synonyms. Why use all these terms instead of one? Because the term request credit is somewhat clumsy, while the term credit does not intuitively get across the idea of a request, while the term request does not intuitively get across the idea of a credit. So, we use all these terms depending on which seems best in the given context.)

The sea is a public directory of requests and is the center of a larger e-mail sending system. We will describe how the sea works in this larger system by examining all the transactions that might be involved in the use of a credit. We also describe the other systems that enable these transactions.

19

The "Life Cycle" of a Request Credit

Request credit transactions can be split into the following stages:

1) creating a request credit listing,
2) searching request listings,
3) sending an L-mail with a credit,
4) redeeming an L-stamp that is affixed together with a credit,
5) complaining about the misuse of a credit,
6) killing a credit.

We will discuss each stage in turn. However, the use of a credit will not necessarily involve all these stages. For example, a credit usually will not be complained about. Whether a credit is involved in a given stage depends on the credit and depends on the parties using the credit.

Note on Guest Credits

The sea can handle guest credits in the same way it handles request credits, except that guest credits entail fewer options and operations. Occasionally we mention guest credits in this section but, as noted, we focus almost exclusively on request credits.

Note on Discounts

We should also note that the sea can include request *discount* information just as it includes request credit information. We restrict the discussion to credits for the sake of simplicity. Most of the discussion applies equally to the use of discounts.

The differences in a credit and discount were explained in Section 3.3. For the sea, these differences mean that a request discount will include a *discount amount* and *discount price* rather than a *credit amount* and a *net price*. Further, request credits can imply bonus amounts. As mentioned, screening e-mail with discounts is harder, in general, than screening e-mail with credits. Thus, we focus on request credits.

The Sea Is Not Tied to L-stamps

We should note the obvious: the information in the sea does not have to refer to payments made by L-stamps. Other kinds of reverse postage schemes are possible. The sea is an advertising medium. It describes how much money people want affixed to e-mails about specified subjects. Of course, the sea differentiates between real money and pseudo-money (credits) but the method of paying the real money is not material.

Later we describe how the sea can incorporate subsidiary functions for transacting L-stamps and registering information about L-stamps. But as a database of advertising information, most of the seas functions are not tied to any particular kind of reverse postage. The L-stamp functions we will discuss are additive and optional.

3.41   Creating a Request Listing

The Sea Briefly Defined

As discussed, the sea is a database system for maintaining request information—receivers (prospective buyers) fill the sea with requests and senders (prospective sellers) fish in the sea for profitable leads.

Bypassing the Sea

Before describing the sea in more detail we should note that a request credit may be created and used independently of the sea. Reece may simply send a credit to Seneca who can then send an L-mail to Reece with that credit affixed. For example, Reece may go to the USAir website which allows her to add her name to a mailing list for fare updates. As a condition of adding her name, she grants USAir permission to affix, say, a $5 credit to each e-mail that USAir sends her. Thus a credit does not have to be administered through the sea. (In case a dispute arises, USAir will have Reece's permission on file and can present it to a judge of some sort who might also deal with credits processed by the sea.)

21

Entering Request Information Into the Sea

Now, presuming that Reece wants to use the sea to advertise her request, the first stage in the life of the request is creating a listing in the sea.

A *request listing* is a set of fields for request information. Not all of the fields will be open to the public. A listing can be thought of in two ways. One, when viewed by users, a listing is like a stock listing on an online database. Two, when viewed by systems operators and by the system itself, it is a file whose information is outputted as the public listing.

So, when we say a *listing* we do not mean just the conventional idea of a public listing but the full set of information that makes a request operational in the sea. We will also refer to a request listing as a *request file* (RF). We will also sometimes use the term *request* or *credit* when referring to a listing. In the sea, these terms all refer to the same entity in memory, the same set of information, that is.

To create a request listing in the sea, Reece enters the information described in the previous section. The sea stores this information in an RF. We give an illustration:

receiver: reece@hotmail.com
subject: hotels in Amalfi with sea views
credit amount: $2
expiration: 11/15/97
signature: PIN 12345

She can enter this information by sending an e-mail to the sea or by using a webform provided by the sea. If she sends in an e-mail, she will have to enter the information in some standardized format so that it can be parsed by the sea. If she uses a webform, it will have the necessary fields for her to enter credit information.

Actually, the only information that Reece has to enter herself is the subject of the request and the credit amount. The sea will automatically capture her address by default if she sends an e-mail. The authentication information may not be necessary either or may be automated. As noted, the boilerplate of a credit grant can be omitted and taken to be understood. Even the expiration date can have a default value.

22

Entering a Net Price or Bonus Amount

Now, if the sea includes a GAP list that contains Reece's GAP then Reece can enter her credit amount in the form of a net price or bonus. For example, let us assume that Reece's GAP is $3. In the case above, she would enter a net price of $1—the GAP ($3) minus the credit amount ($2). For Reece, it may be more convenient to think in terms of a net price to a sender rather than in terms of a credit because she wants to receive the net price as an actual payment. Now, if she grants a credit that is greater than the GAP, she can enter a bonus amount rather than a credit amount. Let us assume that her GAP is $1. In this case, the bonus would be $1. It may be convenient for Reece to think in terms of a bonus because she might think in terms of how much she wants to give above her GAP.

Additional Specifiers

The sea can also include means for enabling Reece to enter the additional request specifiers and conditions described in the previous section. As noted, these would have their own fields in a request listing.

Defaults

The sea can also include means for enabling Reece to set defaults for certain fields. For example, Reece might specify as a default that her requests are to be privacy protected.

Modifying or Canceling a Request

The sea enables Reece to modify or cancel a request. A request file can include password protection enabling only her (and perhaps a system operator) to change information in the file

Having a Surrogate Enter a Credit

The sea can have rules than allow surrogates to enter request information in Reece's stead. Take the illustration of a USAir mailing list. Although Reece could enter the request just as well, she might

23

be too lazy and it might be in the interest of the prospective seller, USAir, to do the work for her. Of course, for this surrogate entering to happen, the surrogate must have some kind of permission from Reece, such as her checking a box at the surrogate's website. For example, as part of signing up on USAir's mailing list, Reece can grant permission to USAir to file a credit with the sea. USAir can perform the entry by automatically sending a form e-mail to the sea when Reece agrees to be put on USAir's mailing list. Thus, a request listing can include a field for designating who has entered the request.

Information Added by the Sea

The sea can include functions for saving Reece steps in entering request information. As noted, if Reece sends her request by e-mail the sea can capture her address from the header of her e-mail, saving her the step of entering the address herself. Since the header address might not be her address it would only be a default address which she could override.

The sea will include functions for adding to the request information Reece enters. This additional information is stored in separate fields. For example, the sea would add the time that a request was entered. Also, the sea would add an ID# for the request. This ID# could be shown to sellers while Reece's e-mail address would not be shown, unless she stipulated that it could be. This ID# is a mechanism for insuring Reece's privacy.

Reece's GAP is also information that the sea can add to a request listing. By extension, the sea can calculate and add the net price or bonus that applies. (If Reece enters a net price or bonus then the sea would calculate and add the credit amount.) These figures can be useful to prospective sellers who will want to know how much they have to pay in real money, if anything, and what kind of credit they are getting relative to Reece's GAP.

So, adding to the illustration above, a request listing might look like:

receiver: reece@hotmail.com
subject: hotels in Amalfi with sea views
credit amount: $2
expiration: 11/15/97
signature: A PIN (not shown but part of a request file)
GAP: $1
Bonus: $1

24

Time entered: 9:30 am, 10/23/97
ID#:   1234512345

Cancellation Field

The sea will also include a public field for indicating whether a request has been canceled before its expiration date. A cancellation field is important because it enables sellers to see whether or not to continue responding to a request. This field is like a "sold" notice posted on a real estate sign.

Additional Request File Information

In Chapter 4, we discuss more information that the sea can register about a request and more information that the sea can add to a request file.

3.42   Searching and Outputting Requests

The purpose of the sea is to enable buyers to advertise requests for sales literature. Sellers search this sea looking for good leads. Vaguely speaking, a seller searches the sea for requests indicating an interest in his product or service. For example, say that our prospective seller, Seneca, is a hotelier in Amalfi. He might then be happy to find Reece's request illustrated in the previous subsection.

More specifically, Seneca wants to find requests that are profitable—where the cost of responding to a request is less than the expected profit generated. To find potentially profitable requests, Seneca must search the sea using a set of criteria and search algorithms provided by himself and/or the sea. For example, he might do a simple keyword search on the phrases *vacations in Amalfi*, *hotels in Amalfi*, and *Amalfi inns*. In his search, he would normally also use financial criteria, e.g., he might say that he only wants to see requests that have a net price of $0 or a bonus.

The sea will return the results of the search and Seneca can then conduct tests to find out what percentage of the requests lead to sales and profits.

25

Subjective Screening

Searching for requests involves subjective screening (matching). A request credit is a grant given by Reece only to senders who abide by the conditions of the grant, especially the condition that they send her e-mails about the product or service she has described in her request. This condition is subjective, of course.

Previously when we have discussed e-mail screening we have been referring to objective screens such as GAP screens. Screening objectively is like having bouncer at a bar who checks a person's ID, while screening subjectively is like having the bouncer check to see if a person is well dressed. Obviously, disputes will arise.

For now, we ignore the potential for disputes and concentrate on the functions of the sea. We assume that there will be meta rules for dealing with senders who send e-mail that poorly matches—or does not match at all—the product or service that Reece specifies in her request. At the end of this subsection we will discuss the matching process further. Right now, we simply assume that senders try to match honestly.

A Public Database

The sea can be a private database but we will assume that it is public. The sea's listings are searched by sellers looking for a list of prospects to send literature to.

The sea can include means for charging users for posting requests, for finding requests, for connect time, etc.. Indeed, the sea needs to be paid for its services. For our purposes, though, we will ignore that aspect, except to note that various charging schemes can be implemented within the sea.

As an online database, the sea will include its own search functions. In addition, the sea can include means for enabling sellers to enter their own their own programs—their own "agents"—for searching the database. In other words, Seneca's search agent could "live" in the sea, continually fishing for profitable listings.

26

So, Seneca uses the sea by entering search criteria and getting a set of matching listings back. Alternatively, he can enter his own search agent which will continually return a set of listings, using the output means of the sea. Alternatively, he can download a partially filtered set of listings from the sea to his own computer and then refine his search there.

The search criteria can, of course, be any of the public fields of information described in Section 3.3. The most important criteria are the keywords or phrases that are meant to match the descriptions in the subject fields of requests. Extra criteria—a product price, a net price, a zip code—can be added to these search parameters. For example, the owner of health club might enter the subject: *health club*, and then narrow the search with a zip code specifier. The zip code can enable him to match all the requests for information about health clubs where the requestors have included a zip code that matches his.

The goal of a search is to enable a seller to find satisfactory matches in the sea of requests. Thus, sellers perform much the same process as buyers. Presuming a seller has something the buyer wants, both parties are playing a match game in which they are trying to locate each other. The buyer and seller have matching interests, so to speak, and thus, both enter much the same information. The buyer enters a set of request information while the seller tries to match pieces of that set.

We have nothing to say about search techniques per se. Particular search methods are beyond the scope of this application. The novelty of the sea resides in the kinds of information that it registers and outputs, and in certain specialized operations relevant to that information (as examples of a special operations, see just below).

Implementing Reece's Instructions

The sea is not just a passive repository of data. The sea also follows Reece's instructions that she includes with her request listing. Certain fields in a request listing are not just conditions for Seneca to follow but instructions for the sea to follow.

If Reece specifies an expiration time/date in a request, the sea hides the request—no longer makes it available to sellers after a that time/date.

If Reece specifies that her request is to be private then the sea does not output Reece's address to Seneca but only the ID# of the request. The ID# can then be submitted by Seneca to a trusted third party (possibly the sea itself) who can convert the ID# to her address in order to send a response from Seneca to her.

If Reece specifies that the request is to be erased completely by a certain time/date then the sea permanently erases the request at that date.

As mentioned previously, the sea does not output a whole request file, only the information that is relevant to Seneca's search and only the information that Reece permits.

Digression on Matching

To see that requests for literature will, like all requests, be subject to interpretation, let us take the subject in the previous subsection: *hotels in Amalfi with sea views.*

Now, say that Seneca is a hotelier in Amalfi but that his hotel, while well appointed and cheap and within a block of the sea, does not have sea views. He might like to send Reece some literature on his hotel and she might be interested in it. Should he send this kind of literature to Reece?

It takes human judgment to decide based upon the meta-rules of the sea. Of course, even with meta-rues, judges will disagree. That is usually unavoidable where requests, descriptions, are concerned.

Users and administrators of the sea will evolve a set of rules to deal with problems. Enforcement is necessary, so the sea or its adjuncts will include means for filing complaints about senders who have misused request listings. For example, a seller of Italian language courses might see the subject of Reece's request above and want to send her literature. Reece might even be interested in an Italian language course but she clearly did not ask for that literature. So, if the seller uses a request credit when mailing that literature to her, that is a breach she could complain about.

To reduce disputes, the sea can also include means for enabling Reece to designate that parts of the subject of her request must be fulfilled by any sender of e-mail literature. For example, Reece might specify that *sea views* are mandatory in any hotel that uses her credit to send her e-mail advertising literature.

28

The sea can also include an extra field in a request listing where Reece designates that she wants the literature to closely match her request. Of course, this kind of instruction is vague and may not be practical. Although the sea can make use of numerous fields for trying to narrow down unsatisfactory matches and wasted e-mails, all measures run up against the seemingly insuperable obstacle of trying to define a good match.

3.43   Affixing Credits and Sending E-mails With Credits

After Finding Requests

Once Seneca has searched the sea and gotten a list of suitable requests, he will want to send e-mails to the Reece's who entered those requests. The Reece's will have granted differing credit amounts and will have differing net prices and bonuses. Therefore, the amount of credit and the amount of L-stamp money, if any, that he will have to affix to each e-mail will vary.

For illustration's sake, we can think of Seneca as a hotelier in Amalfi who finds, say, 100 requests for information about hotels in Amalfi. Now he wants to send e-mail to the requestors.

Some or all of the requests may have privacy designations. In these cases, Seneca will not have the addresses of the requestors but only the request ID#'s. He will have to submit the ID#'s to a third party who will do the sending for him. (The third party will have to have access to the data linking request ID#'s with their corresponding addresses.) In cases where the requests are public, he can do the sending himself with his own LS.

If a third party does the sending it substitutes addresses for the ID#'s it gets and otherwise follows the same procedures that Seneca would follow in affixing credits. So, in presenting steps for affixing a credit, we ignore whether Seneca or his agent are doing the sending.

How Affixing Credits Differs From Affixing L-stamps

29

Credits are somewhat similar to L-stamps in that they serve the same purpose: to add monetary value to an e-mail so that it can be screened and sorted according to how much money is affixed. A credit can be thought of as pseudo, reverse postage.

Affixing a credit is different than affixing an L-stamp primarily because there is no bet information or bet process involved—no random number generation, no EV relative to a payoff amount— there is only a value figure. A credit can be for $.50 but an L-stamp must be for an EV of $.50. A $.50 L-stamp dictates a random number generation (although perhaps not in the affixing) while the a $.50 credit does not refer to any bet process.

So, affixing a credit is a matter of adding to an e-mail: credit amount, an ID, a boilerplate, and, if necessary, a payer and recipient.

C-mail

A credit can be affixed by itself or along with an L-stamp. If only a credit is affixed to an e-mail, we will refer to that e-mail as *C-mail.* If both a stamp and a credit are affixed, we will stick with the term L-mail.

Affixing a Credit Without an L-stamp

We will take the case of affixing an L-stamp by itself and then the case of affixing both together. For convenience, we will say that an LS affixes a credit, although a server that affixes L-stamps is not necessary to affix credits, of course. The important thing is the steps that the computer must perform. However, we can imagine that, to be most useful, a mailserver that affixed credits would also include means for affixing L-stamps.

As noted, to affix a credit a server must add to an e-mail: a credit amount, a credit ID, and a boilerplate. This set of information, along with the receiver's address, is taken from the request listing. Other information, specified by in the request can be added as well.

The credit amount is added to the header. The boilerplate can be added to the body of the e-mail template or it can be signified by a credential that is added to the header. The credit ID can be added to the body template or the header, depending on the form of the ID.

30

In addition, a credit should also include a *title* which is put in the subject header. The title is the subject of the e-mail and is taken by the LS directly from the subject field of the request listing. The title is a phrase from the request that *identifies the request for Reece*. Thus, when Reece sees an e-mail with a credit amount, she can look at the title and remember whether or not she granted that credit. Further, the title can act as the credit ID.

Affixing a Credit Along With an L-stamp

A credit can be affixed along with an L-stamp. In this case, the LS takes information from an L-mail list record for a given receiver and from a request file for that receiver.

When affixing an L-stamp and a credit, the purpose is to add the credit amount and the EV to arrive at a *total payment*. Thus, the LS adds the two figures and affixes a total payment to the e-mail.

Still, the credit and the L-stamp are separate and need to be distinguished in an e-mail (in particular, the two monetary amounts need to be distinguished). The presentation of the L-stamp and credit information is arbitrary. As an example, the L-stamp and credit credentials can both be put in the subject header as can the EV and the credit amount. The problem with this approach is that the subject header becomes cluttered with monetary information. The solution is to have a separate header for monetary information, which allows room to affix a total payment and to distinguish the L-stamp from the credit.

Below we give a procedure that an LS can execute for affixing an L-stamp and a credit. In this procedure, we assume that the LS implements a rule that the total payment should equal a receiver's GAP. Of course, that rule is not mandatory. Affixing an L-stamp and a credit begins with the request listing. We will further assume that the fields of the request listing include a receiver's GAP. Thus, the LS executes the following steps for affixing a credit and L-stamp to an e-mail template:

—Takes a request file,
—pulls the receiver's address, the credit amount and the GAP,
—checks whether there is a positive net price,

31

if the net price is zero or if there is a bonus, affixes the credit, sends the C-mail to the receiver's address.

if there is a positive net price,

creates an L-stamp record for the receiver,

fills in the payer, boilerplate, template ID, and payoff fields as specified by the system operator,

creates an L-stamp ID,

sets the EV equal to the net price,

decides the result of the stamp,

fills the result into the record,

links the record with the request file,

adds the L-stamp information in the record to the template,

adds the credit information in the request file to the template,

adds the credit amount and the L-stamp amount,

puts the sum in the beginning of the subject header,

sends the L-mail to the receiver's address.

3.44   Verifying Credits

Since Reece might not remember all the credits she has granted, so she might want to sometimes verify that a credit she has received is genuine. Thus, the sea can include means for enabling her forward an e-mail with a credit affixed. The sea can include means for capturing the credit information and then verifying the credit. If the sea does not find such a credit in memory, or if the sea finds that the credit amount is inflated, the sea can send the e-mail to the complaint server (described in 3.45) for further action. (A credit verifying server can be separate from the sea provided it has access to the sea's request files.)

3.45   Complaining About Credits

32

A system for using credits needs a resource where people can complain about the misuse of credits. The system requires a *complaint server* (CS) where Reece can forward L-mails that she thinks contain improperly affixed credits. For example, say Reece grants a credit to senders of information about *Hotels in Amalfi*. And say that Berlitz uses the credit to send her information about Italian language courses. That is a misuse of the credit so she would forward Berlitz's e-mail to the CS.

Her complaint would be that the content of the e-mail did not match the subject of the request credit. Other mismatches can occur between what Reece specified in her request and what she receives in an e-mail. For example, if she might specify that a responding e-mail contain a money back guarantee and then find upon receipt of a response that it does not.

There are other misuses. Seneca may use a credit more times than he was allowed. He may use it after it expires.

Thus, the CS would include means for creating complaint *files* for both senders of credits *and* receivers of credits. When the CS receives a complaint, it checks whether or not a file has been set up for the complaining party (identified by her address) and whether a file has been set up for the party being complained about (identified by his address). The CS then stores the complaints (the forwarded e-mails) and tallies them up.

The CS would output an *investigation flag* after receiving a given number of complaints over a given period of time about a particular sender or receiver. The sender or receiver would then be investigated and potentially penalized. The idea is that complaints would not be investigated unless they reached a threshold.

That's because the expense of investigating individual misuses is too great compared to the injury suffered. Further, there will inevitably be a reasonable number of complaints because the match between a request and a responding e-mail is a matter of judgment. Senders and receivers will disagree in some percentage of cases.

There is another area of dispute that the CS can handle. This area involves the faking of *guest credits*—credits where a particular sender is specified rather than a subject.

When Reece and Seneca communicate for business purposes, Seneca will naturally want a guest credit from Reece so that his e-mails will get through her GAP screen. Rather than report every

33

guest credit to the sea, Seneca might just keep Reece's "permission letter" on file in case of dispute. A permission letter can be filled out automatically by Reece, as discussed in 3.41. But, if there is no solid authentication means then Seneca can cheat and claim that Reece gave him a guest credit when she did not. In this case, Reece can complain to the CS which basically follows the same procedure as above. If the CS receives too many complaints about Seneca it can output a flag that will alert a third party to investigate.

A permission letter can be signed with undeniable/unforgeable authentication means but, at present, these means are not widespread. Until these means are widespread, a more practical system may be to omit authentication and rely on the fact that most parties will use guest credits honestly. The exceptional repeat offenders will be caught by the level of complaints against them.

Redeeming Credits

Credits are generally not redeemable because they are pseudo-money. So, when an RS receives an L-mail for redemption that includes a credit, the RS ignores the credit information.

However, a boilerplate condition of a credit might state that the credit becomes a real payment if the credit is misused. In this case, a credit *can* be redeemed. It is the role of the CS and its investigators to determine whether a credit has been misused. So, the CS can include means for redeeming credits when investigators find that credits have been misused.

Once the RS ascertains that a credit can be an actual payment, the RS can redeem the credit by the same expected value payment procedure described in Section 1.10.

3.46 Killing Credits

As mentioned in 3.41, the sea enables Reece to cancel a request. The sea kills a request when Reece cancels it or when its expiration time/date arrives, whichever comes first. But, when we say that the request is killed, that does not mean that the request listing is permanently erased from the sea's memory. In fact, when the expiration time/date arrives, the sea just blocks the output of the listing to the public. For purposes of public searching then, the request is "dead." But, the listing

34

remains stored by the sea for a period of time in case of later dispute and, more importantly, for use in statistical analyses of requests.

If Reece has specified that the request is to be *erased* then the sea could still keep the request stored or it could erase the request. The point of an erasure instruction is that Reece does not want anyone to potentially see that the request is associated with her. Thus, instead of erasing the request, the sea can erase all the information that associates the request with Reece. Other information that is useful for analysis could be kept.

3.47   The Role of the Sea Revisited

In the previous subsections we have explained the various resources that are required for a system that enables people to use request credits (and discounts). The sea is at the center of such a system. As the center, it can have various roles. Most simply, it is a database. Mailing functions can be added to its core database functions so that it become a central mailserver as well—a conduit between buyers and advertisers. Added to its mailing functions can be L-stamp system functions for affixing and redeeming L-stamps. Added to these functions can be the complaint server functions discussed in this section. In other words, the sea can comprise all the functionality described in this specification.

We will briefly illustrate the sea in three roles.

The Sea As Just a Database Hub

Figure 1 shows the sea as purely a data hub for request information. We have our two parties, Seneca and Reece, and we have three servers: the sea, the RS and the CS. We picture a sequence of events. The dashed lines signify events the might or might not take place.

—Reece enters 1 request information into the sea.
—The sea creates 2 a request listing.
—Seneca enters 3 search criteria for requests.
—The sea looks 4 for requests that match of those criteria.

35

--The sea returns **5** a set of matching requests to Seneca.

--Seneca sends **6** an L-mail to Reece with a credit affixed.

Reece can send **7.1** the L-mail to the RS if the L-mail contains an L-stamp. And/or Reece can send **7.2** a complaint to the complain server. And/or Reece can send **7.3** a reply to Seneca.

--If Reece redeems an L-stamp and if Reece wins, the RS notifies **8.1** Seneca. If Reece sends a complaint, and if the level of complaints for Reece or Seneca is greater than a threshold, the CS checks **8.2** with the sea to get the request files corresponding to the complaints. (To reduce clutter in the figure, we omit steps where money is transferred and where the sea responds to the CS.)

The Sea As a Mailing Middleman

In addition to having database functions, the sea can include functions for sending e-mails on behalf of Seneca. In fact, because of privacy issues, the sea or another middleman will have to handle the sending of some (perhaps very large) percentage of e-mails that have credits affixed.

In order to use the sea for mailing his e-mails, Seneca would submit a list of requests to the sea and a template to be sent to the requestors. The sea would then send the template with the corresponding credits affixed. (The sea can also include LS functions for affixing L-stamps along with credits.)

If the sea acts as a mail server, it can also do certain mechanical checks to insure that an e-mail matches the credit affixed to it. Not all the fields of a request can be used this way, only certain ones can. As examples: if a request specifies the format of an e-mail, a type of standard guarantee or a price range then the sea can check that these instructions have been followed by Seneca. The sea checks the information in a request against the information in an e-mail to be sent. In certain cases, the information in the e-mail needs to be placed in standard positions in order for the sea to parse the e-mail to find the information. As another example, the sea can keep a tally of how many times Seneca has used a credit, and if the tally equals the limit set by Reece, the sea can reject the use of the credit. The point is that the sea can include means for following Reece's request instructions rather than relying on Seneca to abide by those instructions.

Figure 2 depicts a sequence of events illustrating the sea's role as a database and mail server:

--Reece enters **21** request information into the sea.

--The sea creates **22** a request listing.

--Seneca enters **23** search criteria for requests.

36

--The sea looks 24 for requests that match of those criteria.

--The sea returns 25 to Seneca a set of ID#'s of matching requests.

--Seneca submits 26 the ID# for Reece's request and submits an e-mail template for the sea to send to Reece, and specifies an L-stamp amount.

--The sea finds the address (Reece's) that correspond to the ID#, and affixes the L-stamp and the credit amount of the request to the template, and sends 27 Reece the L-mail.

The Sea Incorporating All the Functions Discussed

We have split out various functions that need to be performed by a system that processes request credits. As noted, these functions can be performed by separate servers. It would seem more efficient for the sea to incorporate all these functions within one machine or an integrated set of machines. Figure 3 depicts a sequence of event illustrating the sea in the combined roles of database, post office and complaint bureau.

--Reece enters 31 request information into the sea.

--The sea creates 32 a request listing.

--Seneca enters 33 search criteria for requests.

--The sea looks 34 for requests that match of those criteria.

--The sea returns 35 to Seneca a set of ID#'s of matching requests.

--Seneca submits 36 the ID# for Reece's request and submits an e-mail template for the sea to send to Reece, and specifies an L-stamp amount.

--The sea finds the address (Reece's) that corresponds to the ID#, affixes the L-stamp and the credit amount of the request to the template, and sends 37 Reece the L-mail.

--Reece can reply 38.1 to Seneca and/or send 38.2 the sea a redemption e-mail or a complaint about the credit to the sea.

--If Reece redeems the L-stamp, the sea can send 39 Seneca notification.

Chapter 4
Statistical Analyzer of the Value of Requests for E-mail Literature

Is a Request Worth Responding To?

In the previous chapter we described the sea of requests—a specialized database of requests for sales literature. Prospective buyers enter these requests while prospective sellers search through the requests looking for ones that match their products or services. Sellers can then respond to those requests that seem to be good matches.

But is a matching request worth responding to? That is the question that a seller must answer. A response carries a cost: the reverse postage, if any, paid to the receiver plus other expenses associated with creating and sending a message. A seller may also have to consider the potential cost of a series of messages, e-mail and non-e-mail, that may be required to deal with a prospect who shows interest.

Thus, a seller who sends an e-mail ad is a gambler who is betting the cost of the ad that he will make a sale. Usually he can determine his cost and profit. So he needs to know his odds of success. In other words, he can look at a request as a gambling game—slot machine, a lottery ticket, a bingo card. He knows the cost of the bet. He knows what the game will pay off if he wins. But to evaluate whether playing the game is a good bet he must also know the odds of winning.

How the Sea Can Help Estimate the Odds

Yet likening the "playing of a request" to a casino game is not quite right because the odds in a casino game can be determined mathematically. By contrast, requests fall into the category of sports bets and insurance bets—the category of real world gambles where odds are subjective.

Subjective odds are estimated by looking at similar situations in the past. Of course, no one can define "similar situation" so statisticians try to gather data that captures the idea of "similar" with as much detail as possible for a given situation. For example, an insurance company will gather as much data as it can about a driver and then, based on the records of "similar" drivers, will estimate his odds of having an accident.

38

Where requests are concerned, batting statistics provide a good analogy. To judge a batter in baseball, the idea is to collect data about how the batter has hit in all his batting situations. Then statistics can be generated about how likely it is that a batter will get a hit in a given situation—such as "with men on base" or "versus a lefty" or "in a playoff game"—based on how the batter has performed in similar situations in the past. Likewise, the idea is to gather data on how a prospect (Reece) has reacted in all her request situations. That way a request in the present can be evaluated in light of what the prospect has done in similar situations in the past.

In order to evaluate whether an e-mail will spur a sale from Reece, it is necessary to gather data about the e-mail responses that have been sent to her and about her reactions to those responses. That way the sea can generate statistics expressing how her request statements have compared with her buying behavior. The sea can create personal statistics (individual batter type statistics) about her and population statistics (insurance type statistics) about large numbers of receivers.

So, the point of this chapter is mainly that the sea can include means for registering data about responses to requests and about the reactions to those responses.

We define a *response* as an e-mail that is sent by a seller in response to a request.

We define a *reply* as an e-mail message asking for more information.

A *reaction* can mean various things but usually we will mean it as a sale or a reply that is caused by a response.

We use the term *request result* to encompass both responses and reactions to a response. Thus the sea can be a database of requests *and* a database of request results. The sea can also include functions for analyzing result data. If the sea includes such functions we can also think of it as a *statistical request analyzer.*

Request Result Files

The sea stores information about results in *request result files* (RRF's). Each RRF is devoted to a single request.

39

An RRF is a list of records, a table. Each record (each row in the table) is devoted to a single response to a request. The fields in a record then correspond to pieces of data that the sea registers about the response. A record also includes fields for the reaction, if any, to a response.

An RRF can have any number of records (rows) depending, of course, on how many responses there are to a request.

An RRF can be linked to its corresponding request file (RF) or the RRF can be incorporated into its corresponding RF.

The information in Reece's RRF's can be compiled and analyzed to generate statistics about how Reece will react to a response. RRF's from different receivers can be used to generate population statistics.

We cannot, of course, list all the useful data that could be gathered about responses and reactions. We can only give a partial list of certain important kinds of data that the sea can register.

4.1   Response Data That Is Collected

Registering Data About Responses

The sea can register the following data about a response to a request:

a) The identity of the seller.

b) The product or service being sold.

c) The time of the response

d) The L-stamp amount, if any, of the response.

e) The credit amount, if any, of the response.

40

f) The length of the response.

g) The format of the response (text, AVI, GIF, or multi-media file).

h) The price, if any, of the product or service being sold.

i) The type of guarantee, if any, included in the response.

Registering Data About Reactions to Responses

Reece's reaction to a response may be anything from nothing, to a reply asking for more information, to a sale. Actually, reactions can be classified in innumerable ways. We only spell out a partial list of important kinds of data that the sea can register about reactions to a response:

a) A reply to a response.
The sea can also register whether the reply leads to a sale or not.

b) A sale resulting from a response.
The sea can also register information characterizing the communication cost (the effort) involved in making the sale.

c) The amount of a sale resulting from a response.

d) A complaint.
Reece may file a complaint about an L-mail based on the fact that it does not match her request. This complaint can be registered with the sea, and is considered a reaction.

e) A window-shop.
The sea can register that a receiver has replied to a response and engaged the seller in a dialogue resulting in no sale (we call this reaction a *window-shop*). The sea can also register information characterizing the communication cost (the effort) involved in carrying on the dialogue.

f) A welsh.

41

In cases where Reece has made a commitment to buy—in particular, a strike price commitment—Reece may welsh. The sea can register a welsh.

g) A fulfillment.

Reece may fulfill a commitment to buy. The sea can register this fact, just as it can register a welsh.

Results of Ads Sent Over GAP's

Not all e-mail ads will be responses to requests. Ads will also be sent unsolicited over GAP screens. Just as the sea can register data about responses to requests, it can register data about e-mails sent over GAP's. The sea can register this data in *GAP result files* which are analogous to RRF's except that data is not linked to particular request but to a GAP for a particular person. The sea can register the same kind of data about e-mail ads sent unsolicited over GAP's as it can about e-mail ads sent as responses to requests. Naturally, GAP result data can be useful to sellers who want to send unsolicited e-mail ads to receivers with GAP screens.

Storing and Using Sender and Receiver Profiles

Apart from any particular request or response, the sea can input and store data describing senders and receivers—in other words, the sea can input sender and receiver *profiles*. The data in these profiles can then be used to create statistics for estimating the chance that an ad from Seneca will generate a sale or a reply from Reece. A profile may include a wide spectrum of data about an individual and the organization, if any, that he represents. Of course, organizations can be senders and receivers, and they can be profiled too.

Aside: Measuring the Power of a Message

The data that the sea gathers about responses does not concern the quality of a response, except for simple descriptions such as the length of the response or whether the response contains guarantees. Although the "power" of a message is crucial in judging the chances that the message will lead to a

42

sale or a reply, this quality cannot be measured easily. Therefore, the sea's statistics cannot take into account the power of a message. It is up to Seneca to judge how convincing his message is.

Aside: Measuring the Closeness of a Request-Response Match

Another thing that the sea cannot measure is how well a response matches a request. When Reece enters a request she is looking for responses that will closely match what she has requested. But, as noted, advertisers may send her information she wasn't looking for. For example, if Reece requests information about hotels in Ocean City then a hotelier in nearby Bethany Beach might send her information.

Presumably, the closer the match the better the chances of a sale. But we do not know how to measure "closeness" of a match well. Generally, it is up to Seneca to judge how well his response matches Reece's request. But if he gets large numbers of requests from the sea it becomes impractical for him to personally review each request.

Seneca searches the sea for profitable requests by entering various search parameters, especially keyword phrases that match the kind of product or service he sells. For example, he might enter *Bethany Beach hotels, Delmarva hotels, Delaware shore hotels, Maryland shore hotels*, and so on. There are at least three automated approaches that Seneca can use to protect himself from wasting his money on requests that do not match his sales literature.

First, if he is considering using a loose interpretation of a match then he will want to protect himself by specifying a rejection rate for each receiver. As discussed, a rejection is a complaint by Reece that a response does not match her request. A rejection can cost a Seneca money in wasted postage and possibly penalties as well. Thus, Seneca can specify in his search that the requests must come from Reece's who have rejection rates below a certain threshold.

A second approach is to use the matching algorithms of the sea (or whatever search agent Seneca is using) to judge the closeness of a match. The search algorithms internally will generate some kind of match scores. Seneca can use these scores as surrogates for how well his sale literature will match a request. In other words, as part of his search, Seneca can specify "close match" or some such designation that corresponds to a match score level used by the search algorithms. This approach is flawed because search algorithms are not very good at capturing the idea of a match in

43

human terms. Further, Seneca's sales literature may not be well reflected by the search parameters he enters.

A third approach seems best in general. It is to try certain search parameters, take a sample of the requests that are found by using those parameters, and then test that sample. This approach should reveal good search parameters and insulate Seneca from major waste. Further, this approach can be automated because the sea can include means for automatically sending test mails to a sample of the requests that are outputted due to a given set of parameters.

Aside: Seller Questions

We should note that a seller might want to ask Reece to elaborate on what she wants because her request is too vague. We consider a question by a seller as a kind of response. We have chosen not to differentiate responses into categories, although that is certainly possible.

Having said that, we do note one generic response question that the sea can enable sellers to send at reduced cost. It is a question along the lines of *can you be more specific?*.

Aside: Redemption Statistics and Adjusted L-Mail Costs

When Seneca sends an L-mail, he might not have to pay the cost of the L-stamp. That's because when Reece has a choice of redeeming an L-stamp she may choose not to redeem it. So, when Reece has a redemption choice, the L-mail has a discounted expected cost to Seneca.

The sea can include means for registering data and generating statistics about Reece's record of redeeming L-stamps. If the sea includes an RS then the sea can check its own memory to get the information registered by the RS. Otherwise, the sea must include means for downloading the information automatically from an independent RS.

The sea can also include means for converting redemption statistics into an adjusted cost for sending an L-mail response to Reece.

Redemption behavior may correlate with buying behavior so redemption statistics can also be used to evaluate Reece's chances of buying.

44

4.2   How the Sea Can Collect Request Result Data

Request result data can be collected in three general ways:
1) The sea can register result data as part of the process of sending L-mail responses and receiving replies (if the sea sends responses, of course),
2) Senders can report result data to the sea,
3) Receivers can report result data to the sea.

(Note, an RRF will also include a field for recording who reported the result data, if any was reported.)

If the sea acts as an advertising middleman and sends responses, it can also receive replies to those responses. If the sea plays this role it can automatically register responses and replies at least, and perhaps sales. Sales will usually have to be reported by sellers or receivers because sales will usually be transacted outside the sea—the sea being mainly an advertising medium.

The problem with having sellers and receivers do the reporting is that they will not do complete reporting and may not be reliable. Rules can be created for encouraging the reporting of results. For example, Reece may be obligated in some way to report a purchase if she buys a product described in one of her requests. As discussed, sellers may share result data—just as credit card companies share data—to create a cooperative database that reduces losses from wasted advertising.

Reporting by sellers and receivers may or may not yield statistically useful data. The point here is simply that the sea can include mans for enabling sellers and receivers to report request results.

The sea can register results while keeping them partially confidential. For example, if a seller reports a sale, the sea does not have to reveal who the seller was. (It is helpful for a seller to report a sale because that way other sellers will stop sending the buyer responses.)

The sea can also include means for automatically conducting e-mail polls of receivers and sellers to check the results of given requests. For example, the sea can randomly select requests and then send the requestors a query asking whether they have bought as a result of the responses to those

45

requests. In this way data can be gathered on a sampling basis which enables useful population statistics, at least, to be generated.

4.4   Seeing the Competition

If the sea registers request results it can enable sellers to see their competition. This capability has great value because it addresses an enormous problem in the economy: duplication of sales efforts due to ignorance about the competition. Sellers often vie unprofitably for the same buyer simply because they do not know about each other's efforts.

The sea can include functions for outputting RRF information to Seneca that enables him to see how many sellers he is competing against, when they responded to a request, and possibly, who they are. Furthermore, if the sea acts as a post office that handles responses to requests, it can show him the response literature of his competition. (Naturally, policies can vary concerning what information Seneca can see.)

Armed with specific information about who he is competing against, Seneca can make a better judgment about whether it is worthwhile to respond to a request. Moreover, the sea can enable him to search requests according to the level of responses to those requests. That way he can avoid overly competitive situations.

Of course, Seneca may have no chance of winning a sales competition because the competition may be over—Reece may already have bought. Thus, it may be incumbent upon sellers (and/or receivers) to report sales to the sea in order to prevent other sellers from wasting their sales efforts.

Aside: Cheating by Recipients

Reece can cheat Seneca by entering request for a product while having no intention of buying from him. She may indeed buy the product described in the request but, she may also have already decided on the seller before entering her request. Thus, her request is simply a means to get reverse postage from Seneca. This cheat is hard to stop but may not be much of a threat to Seneca as long as the request's net price is not too high.

46

4.5 Using the Result Data

Search According Reaction Criteria

The sea can include search functions that enable Seneca to search the sea using reaction criteria in addition to request criteria. Seneca may want to use reaction criteria in order to test various parameters and see what comes up, or he may have developed his own guidelines as to which characteristics to look for and which to avoid. For example, he may have found it unprofitable to deal with Reece's who have a certain rate of welshing on strike price commitments. Thus, he may narrow his search by specifying requests from Reece's who have never welshed on such commitments. To take another example, he may have found it unprofitable to send responses to people who have already received more than ten responses, so he can narrow his search to requests that have had ten responses or less.

Searching According to the Probability of Success

As mentioned, the sea can include functions for evaluating the probability that Reece will reply to a response or will buy because of a response. We will refer to these combined functions as the sea's *request analyzer* (RA).

The basic procedure of the RA is as follows:

1. Seneca enters search criteria for finding requests.

2. The sea finds and returns a request which we will call the *current request* (normally the sea will return a set of requests but that is irrelevant here).

3. The sea finds a set of requests that are similar to the current request.

4. The sea takes the set of similar requests and analyzes the result data for those requests.

47

5. The sea then outputs the average frequency of each kind of reaction relative to the number of responses. For example, there might be, on average, one sale for every 50 responses to a similar request—a 2% success rate on average.

What do we mean by a "similar" request? To repeat, we cannot define similar. The simplest illustration is to take a single parameter, such as product price, and check the history of other requests with the same instance of that characteristic—in this case, with the same product price. A single parameter approach is usually too crude, just as it is too crude for an insurance company to set rates based upon a single factor such as the age of a driver. Finding similarity usually involves looking at many variables.

As noted, the RA's similarity finding algorithms examine Reece's RF's and RRF's and can examine people's files as well. Generally speaking, individual statstics should be more accurate than population statistics but not always. An individual often will not have provided enough historical data, leaving population statistics as the best alternative.

As with search algorithms, we cannot delve into algorithms that find similarities in data sets. We can say that those skilled in the art will know algorithms to use in the sea, that such algorithms will yield widely varying conclusions, and that many algorithms await discovery.

Single Parameter Illustrations

The RA can help Seneca answer statistical questions such as, *What is the probability of a sale or a reply if I respond to Reece's request given*:
--the number of other responses already sent?
--the time that has passed since the request was entered?
--the price of my product?
--that my response will contains graphics?

As noted, none of the "probabilities" that the RA generates are true probabilities; they are just statistics that summarize past behavior. Although single parameter similarities are not usually adequate, we give a few more examples to give a flavor of the kind of statistics the RA can generate about a request (about Reece's past reactions to responses).

1. The percentage of times Reece buys a product in a request.

48

Let us imagine that Reece has entered 100 requests and that she has purchased products described in 30 of those requests from sellers who responded to those requests. We might then say that there is about a 30% chance that Reece will buy a product from a seller who sends a response to one of her requests.

2. The percentage of times that Reece buys a product in a request that also includes an "I want to buy" message.

When Reece adds an "I want to buy" message she is saying explicitly that she intends to buy. Of course, she may not buy. The RA can generate statistics indicating the predictive value of her "I want to buy" messages.

3. The percentage of times that Reece buys a product in a request given the number of other e-mails sent over the request.

The statistics above ignore the competition that an e-mail ad might face. Competition cannot be measured in the sense of the quality of messages but it can be measured in terms of quantity—the number of responses sent. For example, assume that Reece buys a product described in a request, 30% of the time, and assume she gets an average of 100 responses to a request, then the chances are .3% that an e-mail will be successful (assuming each e-mail has the same chance (1%) of being a winner).

Outputting the Odds of Success

Having found a set of similar requests, and having performed statistical tests upon these to yield the frequency of past reactions, the RA can output the "probability" that a response to the current request will generate a reply or a sale or any other reaction that the sea characterizes.

Thus, when the sea outputs a request it can also include a set of probabilities—*a set of odds*— along with the request listing information described previously. There will be different odds for different reactions. For fun, we can liken these odds to the ones in the racing form.

To illustrate, let us take the fairly clear-cut example of a strike price commitment (see 3.1). Assume that Seneca finds a request by Reece for a new computer system with certain specs, and that the request has a strike price of $7500. And assume that Seneca can put together such a computer and he wants to know whether it is worthwhile to respond. He can check the probability that Reece will welsh given her strike price commitments in past requests. Further the sea can

49

enable him to "refine" this probability by entering a price range. By that we mean that the sea can enable him to see Reece's probability of welshing on requests with strike prices in which she specified a product price in, say, $5,000-$10,000 range. (For illustration's sake, let us presume that the sea has registered enough data about Reece to yield reliable statistics).

Using Probability Thresholds

The capability of assigning probabilities of success to request implies another capability: the sea can enable Seneca to search for requests using the probability of success as a criteria. In other words, in addition to the other criteria he enters to find requests that match his product or service, Seneca can specify a probability level for getting a reply or a sale from his response. For example, Seneca can narrow his search to requests that have a 1% or greater chance of yielding a reply. (The sea can give probabilities for any kind of reaction that the sea characterizes). As stated at the beginning of this chapter, the ability to use the probability of success as a search parameter is the main goal of registering result data.

Using Expected Profit As a Criteria

If the sea can generate reliable probabilities of a sale, it can also enable Seneca to enter a profit figure and then ask for profitable requests. The RA would check the probability of a sale for a given matching request and then multiply by the profit, and then compare the expected profit to the net price for that request. If the net price is less than the expected profit then the request would be considered profitable for Seneca. Of course, this definition of a profitable request only takes into account the net price (the reverse postage) of sending sales literature; it does not include total selling costs. The RA could also enable Seneca to enter a total cost and compare that to the expected profit.

Using expected profit as a parameter is but one of many search parameter options. As discussed, Seneca's best course of action is to test various combinations of parameters, and then test samples of the requests that are outputted to see if a given combination yields requests that are indeed profitable on average.

50

Addendum 1
Paying Mailhosts

L-stamps can be used to pay mailhosts as well as recipients. An L-mail can include two L-stamps, one for Reece and one for the mailhost who handles her mail. The LS can affix both stamps of course.

If a separate stamp is appended to an L-mail for a mailhost, the L-stamp can include the mailhosts address or it can simply have a generic label indicating that the stamp is for the mailhost of the recipient of the L-mail. The mail host can then capture the information and redeem the L-stamp by sending the L-stamp to the RS (to be redeemed in the same way that L-stamps for L-mail recipients are redeemed). The generic label is enough as long as the mailhost can also include the recipient's information to verify that the L-stamp is genuine.

Alternatively, the LS can tally the L-mails sent to a network over a period of time and then pay the owner of the mailhost and standard amount per L-mail. Tallying L-mails obviates the need for L-stamps for mailhosts.

Addendum 2
Ratings

L-stamps whose results are revealed in the body of the L-mails offer a way to count what number of people who open those L-mails. That's because the percentage of winners is known and we can presume that nearly 100% of the winners will respond in order to collect the winnings.

For example, assume that 10,000 L-mails are sent, and each has a stamp with a 1/100 chance of winning. And assume that 40 people respond, then we can say that roughly 4,000 people have opened the L-mails (since we multiply by the reciprocal of the probability of an L-stamp winning). The RS can thus include means for counting and outputting the percentage of people who have opened their L-mails in a given L-mailing.

51

Addendum 3
Connection to AC

(The following comments are cryptic and are made for the purposes of disclosure and priority.)

A request can include a field or fields for enabling Reece to signify that she wants to join with others in a cooperative buying effort. We do not go into details but just note that the sea lends itself readily to enabling people to form buying cooperatives.

A request can also include a field or fields for enabling Reece to offer to pay Seneca actual money for information. One would hope that a system for paying for information is subsumed into the self-organizing database called AC (described in various patent applications).

The sea can also include programming means that implement the semantic linking methods of AC.

Addendum 4
L-Stamps Systems for Faxes and Physical Mail

Fax L-stamps

We can extend the concept of an L-stamp to fax messages. A fax machine can include functions for affixing (printing) an L-stamp on a fax message so that the receiver of the message can cash the L-stamp.

An L-stamp (LS) fax machine, then, includes functions for enabling users to enter the following information to define an L-stamp:
The L-stamp's ID number, the EV, the pay-off amount, the payer, the receiver and the redemption rules (these can be abbreviated by a symbol).

52

An LS fax machine will also include means for printing this information on an outgoing fax message.

An LS fax machine will also include means for recording all the L-stamps that it has printed (sent), including whether or not the fax messages that the L-stamps were affixed to were received or not.

Fax Machines That Screen According to EV's

A fax machine can include functions for screening faxes that do not contain an L-stamp that has an EV greater than or equal to a specified threshold.

Thus a fax machine can include means for inputting a fax GAP—a threshold EV for any incoming fax. Further, such a fax machine can, and would, include means for: detecting an L-stamp, checking the value of the L-stamp, and if the L-stamp does not have an EV greater than or equal to the inputted fax GAP, stopping the transmission.

Physical L-stamps

The preceding chapters treated L-stamps as contract information affixed to e-mail. We can extend the concept to physical mail, and extend an LS to a machine that affixes L-stamps to physical mail—basically a postage meter for physical L-stamps.

A physical L-stamp has the characteristics of an electronic one. The L-stamp includes the stamp ID number, the EV, the payer, the recipient and (perhaps implicitly) the redemption rules.

A physical L-stamp is not ordinary postage. It can be put on the inside of a letter, rather than just on an envelope.

A postage meter for L-stamps is a machine for printing L-stamps onto paper and keeping records of all the stamps printed.

53

Claims

1. I claim:

An online database system inluding communications, processor and memory means, enabling a user to send a set of data to the database through a form, said form incuding the following fields of data:

--a subject,

--a custom admission price defined by the difference between a general admission price and a credit amount,

--an expiration date,

said online database inlcuding online search means enabling user to find said requests according to subject content and price parameters.
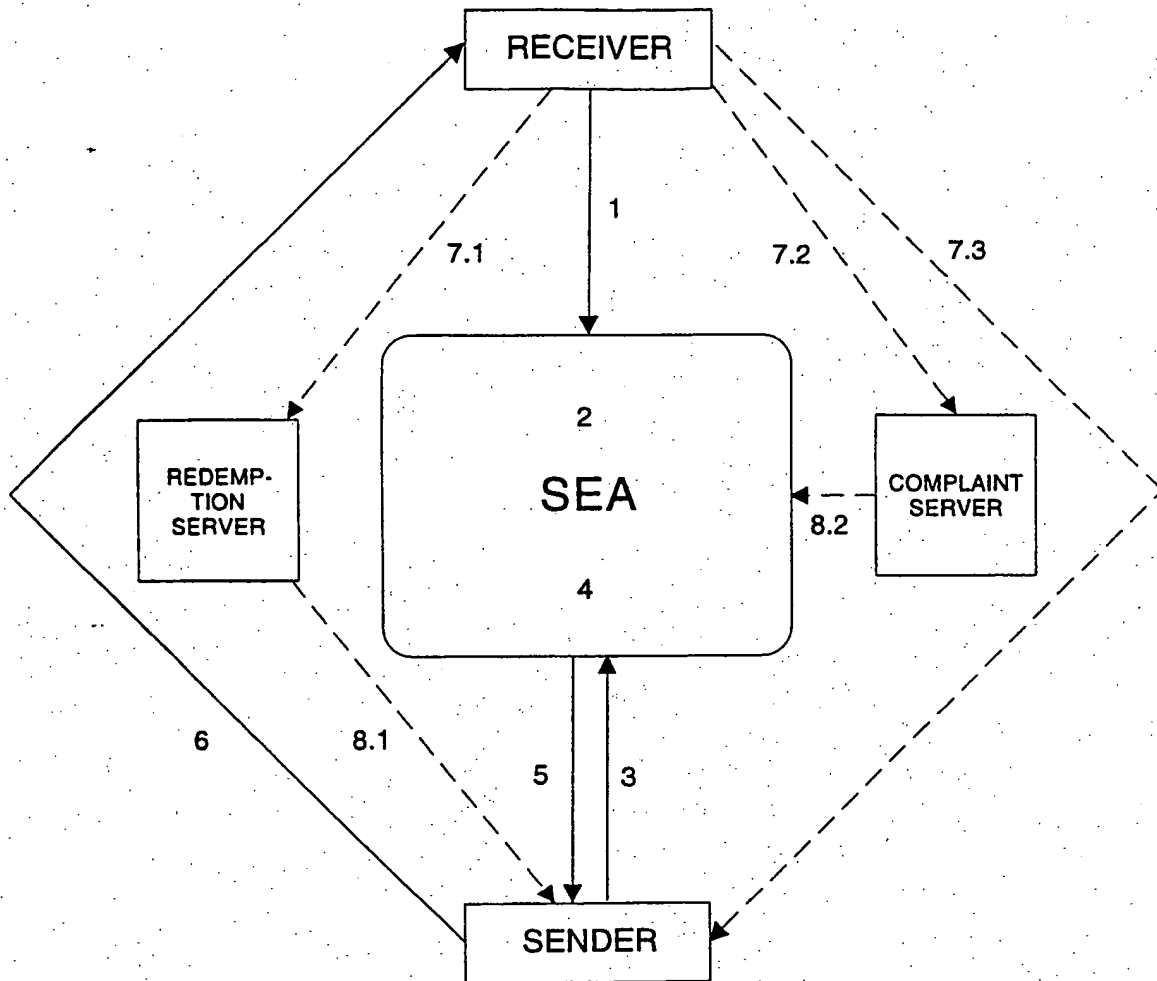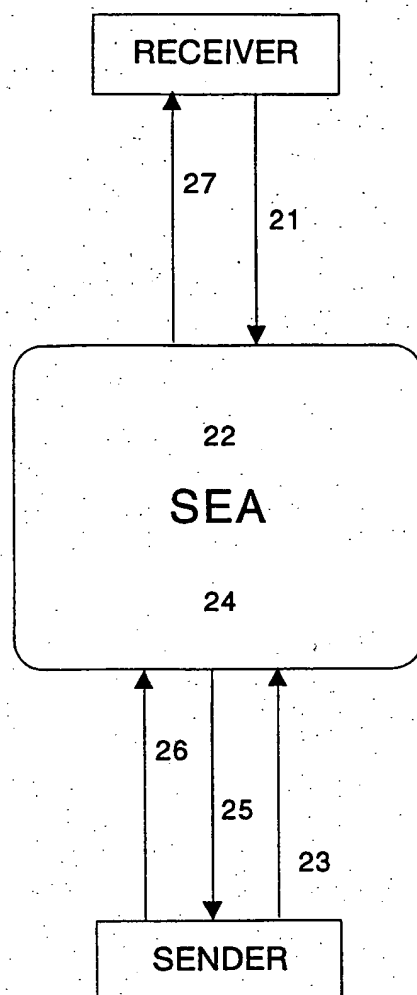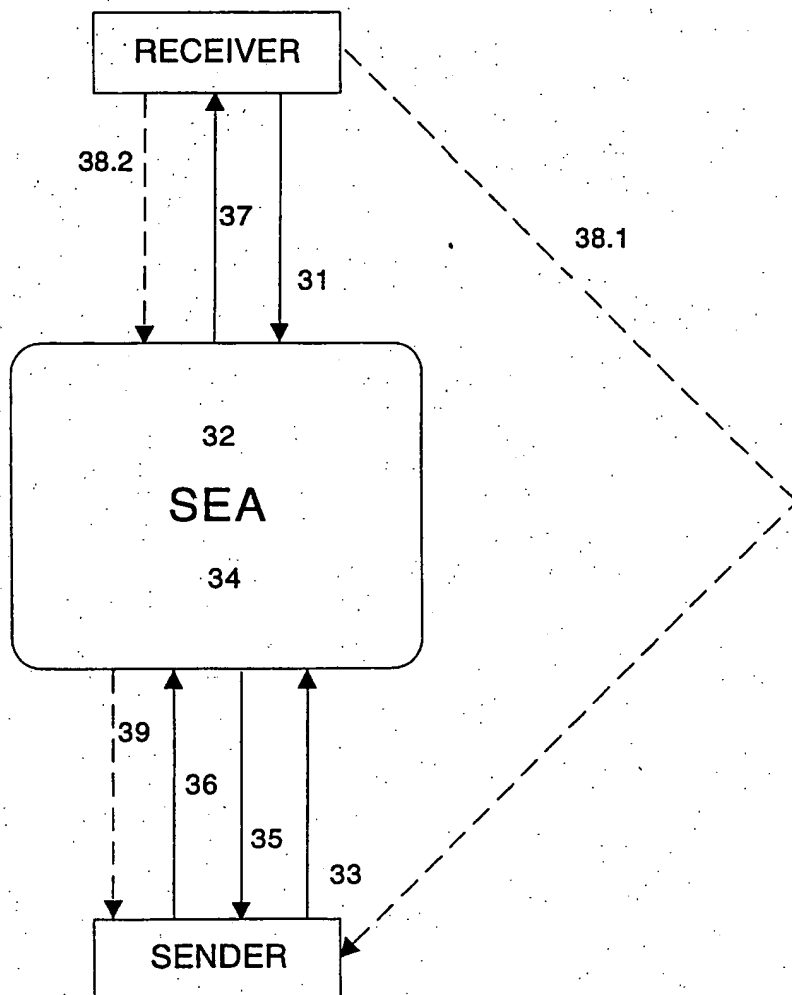
54

FIGURE 1

RECEIVER

27

21

22

SEA

24

26

25

23

SENDER

FIGURE 2

FIGURE 3

THIS PAGE BLANK (USPTO)